**HDU 4183 Pahom on Water**

**Problem Description**

Pahom on Water is an interactive computer game inspired by a short story of Leo Tolstoy about a poor man who, in his lust for land, forfeits everything. The game's starting screen displays a number of circular pads painted with colours from the visible light spectrum. More than one pad may be painted with the same colour (defined by a certain frequency) except for the two colours red and violet. The display contains only one red pad (the lowest frequency of 400 THz) and one violet pad (the highest frequency of 789 THz). A pad may intersect, or even contain another pad with a different colour but never merely touch its boundary. The display also shows a figure representing Pahom standing on the red pad.  
The game's objective is to walk the figure of Pahom from the red pad to the violet pad and return back to the red pad. The walk must observe the following rules:  
1.If pad α and pad β have a common intersection and the frequency of the colour of pad α is strictly smaller than the frequency of the colour of pad β, then Pahom figure can walk from α to β during the walk from the red pad to the violet pad  
2. If pad α and pad β have a common intersection and the frequency of the colour of pad α is strictly greater than the frequency of the colour of pad β, then Pahom figure can walk from α to β during the walk from the violet pad to the red pad  
3. A coloured pad, with the exception of the red pad, disappears from display when the Pahom figure walks away from it.  
The developer of the game has programmed all the whizzbang features of the game. All that is left is to ensure that Pahom has a chance to succeed in each instance of the game (that is, there is at least one valid walk from the red pad to the violet pad and then back again to the red pad.) Your task is to write a program to check whether at least one valid path exists in each instance of the game.

**Input**

The input starts with an integer K (1 <= K <= 50) indicating the number of scenarios on a line by itself. The description for each scenario starts with an integer N (2 <= N <= 300) indicating the number of pads, on a line by itself, followed by N lines that describe the colors, locations and sizes of the N pads. Each line contains the frequency, followed by the x- and y-coordinates of the pad's center and then the radius. The frequency is given as a real value with no more than three decimal places. The coordinates and radius are given, in meters, as integers. All values are separated by a single space. All integer values are in the range of -10,000 to 10,000 inclusive. In each scenario, all frequencies are in the range of 400.0 to 789.0 inclusive. Exactly one pad will have a frequency of “400.0” and exactly one pad will have a frequency of “789.0”.

**Output**

The output for each scenario consists of a single line that contains: Game is VALID, or Game is NOT VALID

**Sample Input**

2

2

400.0 0 0 4

789.0 7 0 2

4

400.0 0 0 4

789.0 7 0 2

500.35 5 0 2

500.32 5 0 3

**Sample Output**

Game is NOT VALID

Game is VALID

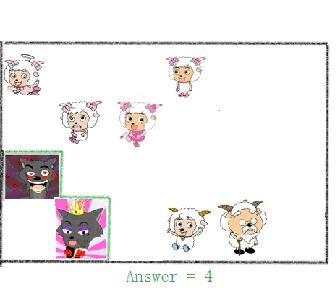
**题目大意：**

有多个点，每个点给出坐标与半径，加入两个点相交，就可以从这两个点走。题目要求先从起点到终点，再从终点回到起点。从起点到终点的过程中，只能从频率小的走到频率大的点（前提是两点相交），从终点到起点的过程中，只能从频率大的走到频率小的。在走的过程中，除了起点与终点，别的只要走过就会消失，就是说只能走一次。问可不可以从起点到终点又回到起点。

**HDU 3046 Pleasant sheep and big big wolf**

**Problem Description**

In ZJNU, there is a well-known prairie. And it attracts pleasant sheep and his companions to have a holiday. Big big wolf and his families know about this, and quietly hid in the big lawn. As ZJNU ACM/ICPC team, we have an obligation to protect pleasant sheep and his companions to free from being disturbed by big big wolf. We decided to build a number of unit fence whose length is 1. Any wolf and sheep can not cross the fence. Of course, one grid can only contain an animal.  
Now, we ask to place the minimum fences to let pleasant sheep and his Companions to free from being disturbed by big big wolf and his companions.



**Input**

There are many cases.   
For every case:   
  
N and M（N,M<=200）  
then N\*M matrix:   
0 is empty, and 1 is pleasant sheep and his companions, 2 is big big wolf and his companions.

**Output**

For every case:  
  
First line output “Case p:”, p is the p-th case;   
The second line is the answer.

**Sample Input**

4 6

1 0 0 1 0 0

0 1 1 0 0 0

2 0 0 0 0 0

0 2 0 1 1 0

**Sample Output**

Case 1:

4

**题目大意**

给一个n\*m的数字阵，1表示羊的位置，2表示狼的位置，0表示没有东西，可以通过。在每个格子的4边都可以建立围栏，有围栏的话狼是不能通过的。目标，建立最少的围栏，将狼和羊完全分开。

**hdu 3452 Bonsai**

**Problem Description**

After being assaulted in the parking lot by Mr. Miyagi following the "All Valley Karate Tournament", John Kreese has come to you for assistance. Help John in his quest for justice by chopping off all the leaves from Mr. Miyagi's bonsai tree!  
You are given an undirected tree (i.e., a connected graph with no cycles), where each edge (i.e., branch) has a nonnegative weight (i.e., thickness). One vertex of the tree has been designated the root of the tree.The remaining vertices of the tree each have unique paths to the root; non-root vertices which are not the successors of any other vertex on a path to the root are known as leaves.Determine the minimum weight set of edges that must be removed so that none of the leaves in the original tree are connected by some path to the root.

**Input**

The input file will contain multiple test cases. Each test case will begin with a line containing a pair of integers n (where 1 <= n <= 1000) and r (where r ∈ ｛1,……, n｝) indicating the number of vertices in the tree and the index of the root vertex, respectively. The next n-1 lines each contain three integers ui vi wi (where ui, vi ∈ {1,……, n} and 0 <= wi <= 1000) indicating that vertex ui is connected to vertex vi by an undirected edge with weight wi. The input file will not contain duplicate edges. The end-of-file is denoted by a single line containing "0 0".

**Output**

For each input test case, print a single integer indicating the minimum total weight of edges that must be deleted in order to ensure that there exists no path from one of the original leaves to the root.

**Sample Input**

15 15

1 2 1

2 3 2

2 5 3

5 6 7

4 6 5

6 7 4

5 15 6

15 10 11

10 13 5

13 14 4

12 13 3

9 10 8

8 9 2

9 11 3

0 0

**Sample Output**

16

**题目大意**

给定一棵树，求最小花费使得根节点与叶子节点断开